Parsimonious Gaussian process models for the classification of hyperspectral remote sensing images

Mathieu Fauvel, Charles Bouveyron, Stéphane Girard

Parsimonious Gaussian process models for the classification of hyperspectral remote sensing images

Mathieu Fauvel, *Member, IEEE,* Charles Bouveyron, and Stéphane Girard

***Abstract*—A family of parsimonious Gaussian process models for classification is proposed in this letter. A subspace assumption is used to build these models in the kernel feature space. By constraining some parameters of the models to be common between classes, parsimony is controlled. Experimental results are given for three real hyperspectral data sets, and comparisons are done with three others classifiers. The proposed models show good results in terms of classification accuracy and processing time.**

***Index Terms*—Kernel methods, remote sensing images, parsimonious Gaussian process, hyperspectral, classification.**

I. Introduction

Thanks to the development of different Earth observation missions, the availability of hyperspectral images with high spatial resolution has increased over the last decade. The fine spectral resolution improves the discrimination of more materials while the high spatial resolution allows the analysis of small structures in the image. Such remote sensing images provide valuable information about landscapes over large areas, on a regular temporal basis and at a relatively low cost. This detailed information is then used in various thematic applications, such as ecological science, urban planing, hydrological science or military applications [1], [2], [3].

One commonly used technique to extract information from remote sensing images is classification [4]. It consists in assigning a label, or a thematic class, to each pixel of the image. Several methods have been developed for images with moderate spectral resolution [5]. However, because of the increasing number of spectral variables in hyperspectral remote sensing images, their classification has become a more and more challenging problem [6]. For instance, model-based classification approaches try to fit the class conditional probability distribution by an ad-hoc parametric model, *e.g.*, Gaussian distribution [4]. However, the estimation of the parameters is difficult when the spectral dimension of the data increases [7], or leads to intractable processing time when non Gaussian models are used. In addition, high spatial resolution images cannot be statistically modeled easily, because spatial details of the image lead to model each class as a mixture of distributions [8]. Alternatively, non-parametric methods such as random forest (RF) classifier have been investigated for the classification of hyperspectral images [9], [10]. However, with the increasing size of the images in the spectral domain, RF requires a sufficient number of training samples to build uncorrelated trees and thus to reach good classification accuracy. This construction is difficult with a reduced sample set and a high number of variables. Unfortunately, the collection of training samples can be a difficult task in remote sensing, resulting in a small training set.

Since the introductory paper in 2005 [11], kernel methods have shown very good abilities in classifying hyperspectral remote sensing images [12]. The use of a kernel function that defines a measure of similarity between two pixel-vectors, makes them robust to the spectral dimension or the non-Gaussianity of the data. The learning step usually involves a constrained optimization problem, where few hyperparameters have to be optimized. Regularization of the decision function is in general included in the optimization problem, making kernel methods robust to the small sample size problem. The support vector machine (SVM) is the most used kernel classifier among the available kernel methods. From its original formulation [13], several methods have been proposed, ranging from the spatial-spectral classification [6] to the semi-supervised classification [12], successfully applied in various domains. The learning step of SVM consists in estimating a separating hyperplane in the kernel feature space, *i.e.*, a linear decision function. In general, most of kernel methods solve a linear problem in the kernel feature space, see for instance kernel Fisher's discriminant analysis or kernel principal component analysis in [12].

Mixing Bayes decision rule and kernel function, M. M. Dundar and A. Landgrebe proposed a kernel quadratic discriminant classifier (KDC) for the analysis of hyperspectral images [14]. It was a first attempt to build a kernel classifier from a quadratic classifier (Gaussian Mixture Model). In order to make the problem tractable, they assumed covariance matrices of all classes to be equal in the kernel feature space. This assumption was proposed by the authors to deal with ill-conditioned kernel matrices. Indeed, unlike the SVM optimization process, no regularization is included in the computation of the decision function with KDC. This function is based on computing the inverse of the centered kernel matrix, which is per construction non-invertible: this $n \times n$ matrix is estimated with the original kernel matrix, from which a combination of rows and lines is removed.

Mathieu Fauvel is with the UMR 1201 DYNAFOR INRA & Institut National Polytechnique de Toulouse, e-mail: mathieu.fauvel@ensat.fr.

Charles Bouveryon is with the Laboratoire MAP5, UMR CNRS 8145, Université Paris Descartes & Sorbonne Paris Cité, e-mail: charles.bouveyron@parisdescartes.fr.

Stéphane Girard is with the Team MISTIS, INRIA Grenoble Rhône-Alpes & LJK, e-mail: Stephane.Girard@inria.fr.

Since then, techniques have been proposed to extend KDC to non-equal covariance matrices. Pseudo inverse and ridge regularization have been proposed in [15]. Xu *et al.* also proposed a KDC where the estimation of the covariance matrix in the feature space is regularized by dropping the smallest eigenvalues from the computation [16]. Similar techniques were used in [17] in the context of small sample size problems.

Although good results in terms of classification accuracy have been reported for the different KDC, several drawbacks can be identified. For instance, [16] and [17] require the estimation of a large number of hyperparameters, while the "equal covariance matrix" assumption in [14] might be too restrictive in practical situations.

In this paper, a family of parsimonious Gaussian process models is reviewed and 5 additional models are proposed to provide more flexibility to the classifier in the context of hyperspectral image analysis. These models allow to build from a finite set of training samples, a Gaussian mixture model in the kernel feature space, where each covariance matrix is free. They assume that the data of each class live in a specific subspace of the kernel feature space. This assumption reduces the number of parameters needed to estimate the decision function and makes the numerical inversion tractable. A closed-form expression is given for the optimal parameters of the decision function. This work extends the models initially proposed in [18], [19]. In particular, the common noise assumption is relaxed, leading to a new set of models for which the level of noise is specific to each class. Furthermore, a closed-form expression for the estimation of the parameters enables a fast estimation of the hyperparameters during the cross-validation step. The contributions of this letter are threefold. 1) The definition of new parsimonious models. 2) A comparison in terms of classification accuracy and processing time of the proposed models with state-of-the-art classifiers of hyperspectral images. 3) A fast cross-validation strategy for learning optimal hyperparameters.

The remainder of the letter is organized as follows. Section II presents the family of parsimonious Gaussian process models as well as the 5 new models. Section III focuses on experimental results obtained on three real hyperspectral data sets. Finally, conclusions and perspectives are discussed in Section IV.

II. Classification with parsimonious Gaussian process models

In this section, it is shown how a Gaussian mixture model (GMM) can be computed in the feature space. It makes use of Gaussian processes as conditional distributions of a latent process. Then estimators are derived and the proposed models are compared to those available in the literature.

A. Gaussian process in the kernel feature space

Let $\mathcal{S} = \left\{(\mathbf{x}_i, y_i)\right\}_{i=1}^{n}$ be a set of training samples, where $\mathbf{x}_i \in \mathbb{R}^d$, is a pixel and $y_i \in \{1, \ldots, C\}$ its class, and C the number of classes. In this work, the Gaussian kernel function is used $k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2_{\mathbb{R}^d}\right)$, with $\gamma > 0$, and its associated mapping function is denoted $\phi : \mathbb{R}^d \to \mathcal{F}$ (the use of another kernel is possible). Its associated feature space $\mathcal{F}$ is infinite dimensional. Therefore the conventional multivariate normal distribution used in GMM cannot be defined in $\mathcal{F}$.

TABLE I
List of sub-models of the parsimonious Gaussian process model.

Model	Variance inside $\mathcal{F}_c$	$\mathbf{q}_{cj}$	p_c
Variance outside $\mathcal{F}_c$: Common			
$p\mathcal{GP}_0$	Free	Free	Free
$p\mathcal{GP}_1$	Free	Free	Common
$p\mathcal{GP}_2$	Common within groups	Free	Free
$p\mathcal{GP}_3$	Common within groups	Free	Common
$p\mathcal{GP}_4$	Common between groups	Free	Common
$p\mathcal{GP}_5$	Common within and between groups	Free	Free
$p\mathcal{GP}_6$	Common within and between groups	Free	Common
Variance outside $\mathcal{F}_c$: Free			
$np\mathcal{GP}_0$	Free	Free	Free
$np\mathcal{GP}_1$	Free	Free	Common
$np\mathcal{GP}_2$	Common within groups	Free	Free
$np\mathcal{GP}_3$	Common within groups	Free	Common
$np\mathcal{GP}_4$	Common between groups	Free	Common

To overcome this, let us assume that $\phi(\mathbf{x})$, conditionally on $y = c$, is a Gaussian process on $J \subset \mathbb{R}$ with mean $\boldsymbol{\mu}_c$ and covariance function $\boldsymbol{\Sigma}_c$. We note $\phi(\mathbf{x})_{cj}$ the projection of $\phi(\mathbf{x})$ on the eigenfunction $\mathbf{q}_{cj}$ using the following scalar product

$$\langle \phi(\mathbf{x}), \mathbf{q}_{cj} \rangle = \int_J \phi(\mathbf{x})(t)\mathbf{q}_{cj}(t)dt.$$

Hence, for all $r \geq 1$, random vectors on $\mathbb{R}^r$ defined by $[\phi(\mathbf{x})_1, \ldots, \phi(\mathbf{x})_r]$ are, conditionally on $y = c$, multivariate normal vectors. In $\mathbb{R}^r$, it is now possible to use the GMM decision rule for class c [20]:

$$D_c\big(\phi(\mathbf{x}_i)\big) = \sum_{j=1}^{r} \left[\frac{\langle \phi(\mathbf{x}_i) - \boldsymbol{\mu}_c, \mathbf{q}_{cj} \rangle^2}{\lambda_{cj}} + \ln(\lambda_{cj}) \right] - 2\ln(\pi_c) \quad (1)$$

where λ_{cj} is the j^{th} eigenvalue of $\boldsymbol{\Sigma}_c$ sorted in decreasing order, $\mathbf{q}_{cj}$ its associated eigenfunction and π_c the prior probability of class c. The classification of $\mathbf{x}_i$ is done to class c if $D_c\big(\phi(\mathbf{x}_i)\big) < D_{c'}\big(\phi(\mathbf{x}_i)\big)$, $\forall c' \in 1, \ldots, C$ [20].

If the Gaussian process is not degenerated (*i.e.*, $\lambda_{cj} \neq 0, \ \forall j$), r has to be large to get a good approximation of the Gaussian process. Unfortunately, only a part of the above equation can be computed from a finite training sample set:

$$D_c(\phi(\mathbf{x}_i)) = \underbrace{\sum_{j=1}^{r_c} \left[\frac{\langle \phi(\mathbf{x}_i) - \boldsymbol{\mu}_c, \mathbf{q}_{cj} \rangle^2}{\lambda_{cj}} + \ln(\lambda_{cj}) \right] - 2\ln(\pi_c)}_{\text{computable quantity}} + \underbrace{\sum_{j=r_c+1}^{r} \left[\frac{\langle \phi(\mathbf{x}_i) - \boldsymbol{\mu}_c, \mathbf{q}_{cj} \rangle^2}{\lambda_{cj}} + \ln(\lambda_{cj}) \right]}_{\text{non computable quantity}}$$

where $r_c = \min(n_c, r)$ and n_c is the number of training samples of class c. Consequently, the decision rule cannot be computed in the feature space if $r > n_c$, for $c = 1, \ldots, C$.

B. Parsimonious Gaussian process

To make the above computational problem tractable, it is proposed to use a parsimonious Gaussian process model in the feature space for each class. These models assume that each class is located in a low-dimensional subspace of the kernel feature space. In [18], it was assumed the noise level is common to all classes (*Definition* 1). In this paper, these models are extended to the situation where the noise level can be dependent on the class (*Definition* 2).

Definition 1 (Parsimonious Gaussian process with common noise): A parsimonious Gaussian process with common noise is a Gaussian process $\phi(\mathbf{x})$ for which, conditionally to $y = c$, the eigen-decomposition of its covariance operator $\mathbf{\Sigma}_c$ is such that

- A1. It exists a dimension $r < +\infty$ such that $\lambda_{cj} = 0$ for $j \geq r$ and for all $c = 1, \ldots, C$.
- A2. It exists a dimension $p_c < \min(r, n_c)$ such that $\lambda_{cj} = \lambda$ for $p_c < j < r$ and for all $c = 1, \ldots, C$.

Definition 2 (Parsimonious Gaussian process with class specific noise): A parsimonious Gaussian process with class specific noise is a Gaussian process $\phi(\mathbf{x})$ for which, conditionally to $y = c$, the eigen-decomposition of its covariance operator $\mathbf{\Sigma}_c$ is such that

- A3. It exists a dimension $r_c < r$ such that $\lambda_{cj} = 0$ for all $j > r_c$ and for all $c = 1, \ldots, C$. When $r = +\infty$, it is assumed that $r_c = n_c - 1$.
- A4. It exists a dimension $p_c < r_c$ such that $\lambda_{cj} = \lambda_c$ for $j > p_c$ and $j \leq r_c$, and for all $c = 1, \ldots, C$.

Assumptions *A1* and *A3* are motivated by the quick decay of the eigenvalues for a Gaussian kernel [21]. Hence, it is possible to find $r < +\infty$ (or r_c) such as $\lambda_{cr} \approx 0$. Assumptions *A2* and *A4* express that the data of each class live in a specific subspace of size p_c, the signal subspace, of the feature space. The variance in the signal subspace for class c is modeled by parameters $\lambda_{c1}, \ldots, \lambda_{cp_c}$ and the variance in the noise subspace is modeled by λ or λ_c. This model is referred to by $p\mathcal{GP}_0$ for the common-noise assumption or $np\mathcal{GP}_0$ for the class-specific noise assumption.

From these models, it is possible to derive several sub-models. Table I lists the different sub-models that can be built from $p\mathcal{GP}_0$ and $np\mathcal{GP}_0$. For models $p\mathcal{GP}_1$ and $np\mathcal{GP}_1$, it is additionally assumed that data of each class share the same intrinsic dimension, *i.e.*, $p_c = p$, $\forall c \in \{1, \ldots, C\}$. In models $p\mathcal{GP}_2$ and $np\mathcal{GP}_2$, variance in the signal subspace $\mathcal{F}_c$ is assumed to be equal for all eigenvectors, *i.e.*, $\lambda_{cj} = \lambda_c$, $\forall j \in \{1, \ldots, p_c\}$. For models $p\mathcal{GP}_4$ and $np\mathcal{GP}_4$, it is assumed that the intrinsic dimension is common to every class and the variance is common between them, *i.e.*, $\lambda_{cj} = \lambda_{c'j}$, $\forall j \in \{1, \ldots, p\}$ and $c, c' \in \{1, \ldots, C\}$. In term of parsimony, $p\mathcal{GP}_0$ is the least parsimonious model while $p\mathcal{GP}_6$ is the most parsimonious one for models with common noise. $p\mathcal{GP}_0$ is also more parsimonious than $np\mathcal{GP}_0$. A visual illustration of $p\mathcal{PGP}_1$ in $\mathbb{R}^2$ is shown in Fig. 1.

In the following, only model $np\mathcal{GP}_0$ is discussed. Similar results can be obtained for other models and a discussion of model $p\mathcal{GP}_0$ can be found in [18], [19].

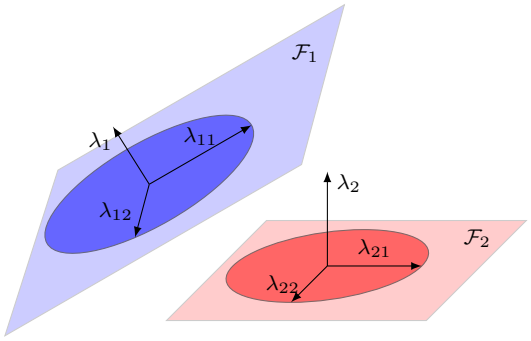

Fig. 1. Visual illustration of model $np\mathcal{GP}_1$. Dimension of $\mathcal{F}_c$ is common to both classes, they have specific variance inside $\mathcal{F}_c$ and they have specific noise level.

Proposition 1: Eq. (1) can be written for $np\mathcal{GP}_0$ as

$$D_c\big(\phi(\mathbf{x}_i)\big) = \sum_{j=1}^{p_c} \frac{(\lambda_c - \lambda_{cj})}{\lambda_{cj}\lambda_c} \langle \phi(\mathbf{x}_i) - \boldsymbol{\mu}_c, \mathbf{q}_{cj} \rangle^2 + \frac{\|\phi(\mathbf{x}_i) - \boldsymbol{\mu}_c\|^2}{\lambda_c} + \sum_{j=1}^{p_c} \ln(\lambda_{cj}) + (r_c - n_c)\ln(\lambda_c) - 2\ln(\pi_c). \tag{2}$$

Computation of eq. (2) is now possible since $p_c < n_c$, $\forall c \in \{1, \ldots, C\}$. In the following, it is shown that the estimation of parameters and the computation of eq. (2) can be done using only kernel evaluations, as in standard kernel methods.

C. Model inference

Centered Gaussian kernel function according to class c is defined as $\bar{k}_c(\mathbf{x}_i, \mathbf{x}_j) = k(\mathbf{x}_i, \mathbf{x}_j) + \frac{1}{n_c^2} \sum_{\substack{l,l'=1 \\ y_l, y_{l'}=c}}^{n_c} k(\mathbf{x}_l, \mathbf{x}_{l'}) - \frac{1}{n_c} \sum_{\substack{l=1 \\ y_l=c}}^{n_c} \big(k(\mathbf{x}_i, \mathbf{x}_l) + k(\mathbf{x}_j, \mathbf{x}_l)\big)$. Its associated normalized kernel matrix $\overline{\mathbf{K}}_c$ of size $n_c \times n_c$ is defined by $(\overline{\mathbf{K}}_c)_{l,l'} = \bar{k}_c(\mathbf{x}_l, \mathbf{x}_{l'})/n_c$. With these notations, the following results hold for $np\mathcal{GP}_0$.

Proposition 2: For $c = 1, \ldots, C$ and under model $np\mathcal{GP}_0$, eq. (2) can be computed with:

$$D_c\big(\phi(\mathbf{x}_i)\big) = \frac{1}{n_c} \sum_{j=1}^{\hat{p}_c} \frac{\hat{\lambda}_c - \hat{\lambda}_{cj}}{\hat{\lambda}_{cj}^2 \hat{\lambda}_c} \Bigg(\sum_{\substack{l=1 \\ y_l=c}}^{n_c} \beta_{cjl} \bar{k}_c(\mathbf{x}_i, \mathbf{x}_l) \Bigg)^2 + \frac{\bar{k}_c(\mathbf{x}_i, \mathbf{x}_i)}{\hat{\lambda}_c} + \sum_{j=1}^{\hat{p}_c} \ln(\hat{\lambda}_{cj}) + (\hat{r}_c - \hat{p}_c)\ln(\hat{\lambda}_c) - 2\ln(\pi_c),$$

where

$$\hat{\lambda}_c = \big(\text{trace}\,(\overline{\mathbf{K}}_c) - \sum_{j=1}^{p_c} \lambda_{cj}\big)/(r_c - p_c).$$

Estimation of p_c is done by looking at the cumulative variance for sub-models $p\mathcal{GP}_{0,2,5}$. In practice, p_c is estimated such as the percentage of the cumulative variance is higher than a given threshold t. For other sub-models, $\hat{p}$ is a fixed parameter given by user. Finally, all parameters can be inferred from $\overline{\mathbf{K}}_c$.

D. Link with existing models

The associated of equal covariance matrices in [14] corresponds to our model $p\mathcal{GP}_4$ with an additional equality constraint on the eigenfunction. By using parsimonious Gaussian process, we are able to provide more flexibility in the feature space by allowing covariance matrix of each class to be different, and for the 5 new models, by allowing the noise in each class to be different. Furthermore, the storage complexity of [14] is $\mathcal{O}(n^2)$, since it works on the full kernel matrix, while the storage complexity of our models is $\mathcal{O}(n_c^2)$, usually very much lower. Furthermore, the eigendecomposition of the kernel matrix is of complexity $\mathcal{O}(n^3)$ for [14], while it is reduced to $\mathcal{O}(n_c^3)$ with our models.

Authors of [14] also implement a ridge regularization to stabilize the generalized eigenvalue problem. Numerically, it is equivalent to set small eigenvalues to a constant term, which is similar to *A2* and *A4* in *Definition* 1 and *Definition* 2. In the same way, KDC models proposed in [15] use ridge regularization, but they are constructed for each class, *i.e*, each class has a separate covariance matrix. The main difference between ridge regularization and our models is that, with ridge regularization, eigenvectors corresponding to very small values are still computed and used in the decision function while our models only use the p_c first ones. Note that KDC was also extended to indefinite kernel functions in [15] by using Moore-Penrose inverse, *i.e.*, eigenvalue thresholding.

Covariance regularization techniques proposed in [22] were used in [16] and [17]. In addition to kernel hyperparameter, two additional hyperparameters have to be tuned. In practice, even for moderate size problem it can be very time consuming. Our models only have two hyperparameters to tune, and one can be computed for a moderate numerical cost, as it is explained in II-E.

The model proposed in [16] is similar to $np\mathcal{GP}_0$. The authors proposed to estimate the p first eigenvalues for each class and set the noise term to the value of the $(p+1)^{\text{th}}$ eigenvalue. In our model, the noise term is estimated as the mean value of the remaining eigenvalues. Additional flexibility is provided by our models since the size of the signal subspace can be class dependent.

E. Estimation of the hyperparameters

For each proposed model, there are two hyperparameters to tune: the scale γ of the Gaussian kernel and the size p_c of $\mathcal{F}_c$ or the percentage of cumulative variance t. In this work, the v-fold cross-validation (CV) strategy is employed. For the last two parameters, it is possible to use a strategy that speed-up the computing time. The most demanding part in terms of processing time of the proposed models is the eigendecomposition of $\overline{\mathbf{K}}_c$. But for a given value of γ and a given fold of the CV, it is possible to compute the eigendecomposition of $\overline{\mathbf{K}}_c$ only once. From the decomposition, all the model parameters for every values of p_c or t are available at not cost since they are derived from the eigenvectors and eigenvalues of $\overline{\mathbf{K}}_c$. It allows efficient computation of the CV error estimate for pairs of hyperparameters.

This fast computation of the CV error is possible because the model parameters are obtained through an explicit formulation, contrary to SVM for which a optimization procedure is required and need to be restarted when a new set of hyperparameters is tested.

III. Experimental results

A. Data sets and benchmarking methods

Three hyperspectral data sets have been used in these experiments.

University of Pavia: The data set has been acquired by the ROSIS sensor during a flight campaign over Pavia, northern Italy. 103 spectral channels were recorded from 430 to 860 nm. 9 classes have been defined for a total of 42,776 referenced pixels.

Kennedy Space Center: The data set has been acquired by the AVIRIS sensor during a flight campaign over the Kennedy Space Center, Florida USA. 224 spectral channels were recorded from 400 to 2500 nm. Because of water absorption the final data set contains 176 spectral bands. 13 classes have been defined for a total of 4,561 referenced pixels.

Heves: The data set has been acquired by the AISA Eagle sensor during a flight campaign over Heves, Hungary. It contains 252 bands ranging from 395 to 975 nm. 16 classes have been defined for a total of 360,953 referenced pixels.

For each data set, 50 training pixels per class were randomly selected and the remaining referenced pixels were used for validation. 20 repetitions were done for which a new training set have been generated randomly. The range of each spectral variable has been stretched between 0 and 1.

Reported results are the average Kappa coefficient and the average processing time in seconds (including selection of hyperparameters, training process and prediction process). In order to test the statistical significance of the observed differences, a Wilcoxon rank-sum test has been computed between each pair of methods.

For comparison, SVM and RF classifiers have been tested using the Scikit-learn Python package [23]. Furthermore, the KDC of [14] has been implemented. The parsimonious models have been implemented in Python, and codes can be download here: https://github.com/mfauvel/PGPDA. All hyperparameters of each method have been selected using a 5-fold cross validation.

B. Discussion

Results are reported in Table II. In terms of accuracy, one of the proposed parsimonious models performs the best for each data set. SVM performs the best for two data sets and KDC performs the best for one data set. In particular, for *Heves* data set, $(n)p\mathcal{GP}_1$ provides the best results. For *University of Pavia* and *Kennedy Spectral Center* data sets, SVM provides the best results but the differences with $(n)p\mathcal{GP}_1$ are not statistically significant.

RF usually provides lower accuracy. However, RF is the fastest algorithm, by far. KDC performs the worst in terms of processing time, because each class involves a $n \times n$ kernel

TABLE II
EXPERIMENTAL RESULTS IN TERMS OF ACCURACY AND PROCESSING TIME. THE VALUES REPORTED CORRESPOND TO THE AVERAGE RESULTS OBTAINED ON 20 REPETITIONS. BOLDFACE RESULTS INDICATE BEST RESULTS FOR A GIVEN DATA SET. MULTIPLE BOLDFACE RESULTS INDICATE THAT DIFFERENCES BETWEEN THEM ARE NOT STATISTICALLY SIGNIFICANT.

	Kappa coefficient			Processing time (s)		
	University	KSC	Heves	University	KSC	Heves
$p\mathcal{GP}_0$	0.768	0.920	0.664	18	31	148
$p\mathcal{GP}_1$	**0.793**	**0.922**	**0.671**	18	33	151
$p\mathcal{GP}_2$	0.617	0.844	0.588	18	31	148
$p\mathcal{GP}_3$	0.603	0.842	0.594	19	33	152
$p\mathcal{GP}_4$	0.661	0.870	0.595	19	34	152
$p\mathcal{GP}_5$	0.567	0.820	0.582	18	32	148
$p\mathcal{GP}_6$	0.610	0.845	0.583	19	34	152
$np\mathcal{GP}_0$	0.730	0.911	0.640	17	31	148
$np\mathcal{GP}_1$	**0.792**	0.921	**0.677**	18	33	151
$np\mathcal{GP}_2$	0.599	0.838	0.573	18	31	148
$np\mathcal{GP}_3$	0.578	0.817	0.585	19	33	152
$np\mathcal{GP}_4$	0.578	0.817	0.585	19	33	152
KDC	0.786	**0.924**	0.666	98	253	695
RF	0.646	0.853	0.585	**3**	**3**	**18**
SVM	**0.799**	**0.928**	0.658	10	28	171

matrix. Parsimonious models perform on average as fast as SVM and are much faster than KDC.

For the proposed models, best results are obtained by $(n)p\mathcal{PG}_1$, which are the least parsimonious models. They have free variance inside the signal subspace but common dimension of subspaces. $(n)p\mathcal{PG}_0$ performs slightly worse than SVM and KDC but better than RF. All the other models are less accurate. There is no difference in terms of processing time between parsimonious models, since they all rely on the eigendecomposition of $\overline{\mathbf{K}}_c$.

IV. CONCLUSIONS

Parsimonious Gaussian process models have been proposed in this letter. They allow the computation of a kernel quadratic discriminant classifier with limited training samples. The main assumption considered in this work is that relevant information for the discrimination task is located in a smaller subspace of the kernel feature space. Sub-models are derived by constraining some properties of the models to be common between classes, thus enforcing the parsimony. Moreover, new models have been discussed for which the noise level is specific to the class.

The proposed models have been compared in terms of classification accuracy and processing time with three other classifiers, on three real hyperspectral data sets. Results show that two proposed models ($(n)p\mathcal{PG}_1$) are very effective both in terms of accuracy and in terms of processing time. However, other proposed models do not provide as good classification accuracy, and would not be appropriate for practical situations. From our experimental results, $p\mathcal{PG}_1$ and $np\mathcal{PG}_1$ behave similarly in terms of computation time or classification accuracy.

The comparison with KDC shows that $(n)p\mathcal{PG}_1$ models are competitive, with better classification accuracy and smaller processing time. They offer a good alternative to the conventional KDC method.

REFERENCES

[1] A. Ghiyamat and H. Shafri, "A review on hyperspectral remote sensing for homogeneous and heterogeneous forest biodiversity assessment," *International Journal of Remote Sensing*, vol. 31, no. 7, pp. 1837–1856, 2010.

[2] P. Hardin and A. Hardin, "Hyperspectral remote sensing of urban areas," *Geography Compass*, vol. 7, no. 1, pp. 7–21, 2013.

[3] X. Briottet, Y. Boucher, A. Dimmeler, A. Malaplate, A. Cini, M. Diani, H. Bekman, P. Schwering, T. Skauli, I. Kasen, I. Renhorn, L. Klasén, M. Gilmore, and D. Oxford, "Military applications of hyperspectral imagery," in *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, June 2006, vol. 6239.

[4] D.A. Landgrebe, *Signal Theory Methods in Multispectral Remote Sensing*, John Wiley and Sons, New Jersey, 2003.

[5] B. Tso and P. M . Mather, *Classification Methods for Remotely Sensed Data, Second Edition*, CRC Press, 209.

[6] M. Fauvel, Y. Tarabalka, J. A. Benediktsson, J. Chanussot, and J. Tilton, "Advances in Spectral-Spatial Classification of Hyperspectral Images," *Proceedings of the IEEE*, vol. 101, no. 3, pp. 652–675, Mar. 2013.

[7] L. O. Jimenez and D. A. Landgrebe, "Supervised classification in high-dimensional space: geometrical, statistical, and asymptotical properties of multivariate data," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 28, no. 1, pp. 39 –54, feb 1998.

[8] A. Berge and A. H. S. Solberg, "Structured gaussian components for hyperspectral image classification," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 44, no. 11, pp. 3386–3396, Nov 2006.

[9] H. Jisoo, C. Yangchi, M. M. Crawford, and J. Ghosh, "Investigation of the random forest framework for classification of hyperspectral data," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 43, no. 3, pp. 492–501, March 2005.

[10] P. O. Gislason, J. A. Benediktsson, and J. R. Sveinsson, "Random forests for land cover classification," *Pattern Recognition Letters*, vol. 27, no. 4, pp. 294 – 300, 2006.

[11] G. Camps-Valls and L. Bruzzone, "Kernel-based methods for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, pp. 1351–1362, 2005.

[12] G. Camps-Valls and L. Bruzzone, Eds., *Kernel Methods for Remote Sensing Data Analysis*, Wiley, 2009.

[13] F. Melgani and L. Bruzzone, "Classification of hyperspectral remote sensing images with support vector machines," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 42, no. 8, pp. 1778–1790, 2004.

[14] M. Dundar and D. A. Landgrebe, "Toward an optimal supervised classifier for the analysis of hyperspectral data," *IEEE Trans. Geoscience and Remote Sensing*, vol. 42, no. 1, pp. 271–277, 2004.

[15] E. Pekalska and B. Haasdonk, "Kernel discriminant analysis for positive definite and indefinite kernels," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 6, pp. 1017–1032, 2009.

[16] Z. Xu, K. Huang, J. Zhu, I. King, and M. R. Lyu, "A novel kernel-based maximum a posteriori classification method," *Neural Netw.*, vol. 22, no. 7, pp. 977–987, Sept. 2009.

[17] J. Wang, K. N. Plataniotis, J. Lu, and A. N. Venetsanopoulos, "Kernel quadratic discriminant analysis for small sample size problem," *Pattern Recognition*, vol. 41, no. 5, pp. 1528 – 1538, 2008.

[18] C. Bouveyron, M. Fauvel, and S. Girard, "Kernel discriminant analysis and clustering with parsimonious Gaussian process models," *Statistics and Computing*, pp. 1–20, 2014.

[19] M. Fauvel, C. Bouveyron, and S. Girard, "Parsimonious gaussian process models for the classification of multivariate remote sensing images," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, May 2014, pp. 2913–2916.

[20] G. J. McLachlan, *Discriminant analysis and statistical pattern recognition*, Wiley series in probability and mathematical statistics. J. Wiley and sons, New York, Chichester, Brisbane, 1992.

[21] M. L. Braun, J. M. Buhmann, and K.-R. Müller, "On relevant dimensions in kernel feature spaces," *J. Mach. Learn. Res.*, vol. 9, pp. 1875–1908, June 2008.

[22] J. H. Friedman, "Regularized discriminant analysis," *Journal of the American Statistical Association*, vol. 84, no. 405, pp. pp. 165–175, 1989.

[23] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.